第 39 卷第 8 期
录用定稿
计算机应用研究
Application Research of Computers
Vol. 39 No. 8
Accepted Paper

基于图注意力网络的开源社区问题解决参与者推荐 *

赵海燕[1, 2, 3]，夏文宗[1, 2, 3†]，曹　健[4]，陈庆奎[1, 2, 3]

(1. 上海市现代光学系统重点实验室，上海 200093; 2. 光学仪器与系统教育部工程研究中心，上海 200093; 3. 上海理工大学 光电信息与计算机工程学院，上海 200093; 4. 上海交通大学 计算机科学与技术系，上海 200030)

摘　要：在开源社区中，开发者提出的问题能否得到快速与高质量的答复和解决决定着社区的活跃程度。因此，为新提交的问题寻找和推荐合适的问题解决参与者有助于社区的发展。根据开发者之间的协作关系记录与开发者参与问题的记录构建了双层图注意力网络的问题解决参与者推荐模型(GAT-UCG)。首先获取问题参与者的信息和开发者的互动信息，分别构建开发者问题参与图和开发者协作关系图，通过注意力机制对于边重新分配权重，最后根据输出层得到的问题节点嵌入表示进行问题参与者的 Top-N 推荐。选取了 Github 流行仓库中的 7352 个问题进行了实验，实验结果表明，所提出的 GAT-UCG 模型在推荐准确率、召回率、F-Score 三个指标上均优于基线方法。

关键词：推荐系统；问题跟踪；图注意力网络；参与者推荐；评论网络
中图分类号：TP183　　doi: 10.19734/j.issn.1001-3695.2022.01.0028

Graph attention network based participant recommendation for issue resolution in open source community

Zhao Haiyan[1, 2, 3], Xia Wenzong[1, 2, 3†], Cao Jian[4], Chen Qingkui[1, 2, 3]

(1. *Shanghai Key Lab of Modern Optical System*, *Shanghai* 200093, *China*; 2. *Engineering Research Center of Optical Instrument & System*, *Ministry of Education*, *Shanghai* 200093, *China*; 3. *School of Optical-Electrical & Computer Engineering*, *University of Shanghai for Science & Technology*, *Shanghai* 200093, *China*; 4. *Dept. of Computer Science & Technology*, *Shanghai Jiao Tong University*, *Shanghai* 200030, *China*)

Abstract: In the Open Source Community, it's essential to find and recommend suitable participants for newly initiated issues in order to solved the issues and develop the community. This paper proposed to construct a two-layer Graph Attention Network Participant Recommendation Model (GAT-UCG) based on the cooperative relationship records and the historical participated issues records of the developers. The method used to construct the model is obtaining the information of the problem participants and the interaction information of the developers first, and building the developer problem participation graph and the developer collaboration relationship graph respectively, then redistributing the weights to the edges through the attention mechanism, finally, figuring the Top-N recommendation of the problem participants according to the Issue node embedding representation obtained by the output layer. There are 7, 352 issues from popular Github repositories for experiments. The results showed that the GAT-UCG model outperforms the baseline method in three indicators: recommendation accuracy, recall, and F-Score.

Key words: recommendation system; issue tracking; graph attention network; participant recommendation; comment network

0　引言

在现代开源软件开发过程中，开源社区为开源软件的开发者们提供了一个高效的合作平台，例如，Github 作为全球最大的开源社区平台，吸引了世界各地的开发者参与开发。

为了让开发者更便捷的进行开源项目的交流，每个人都可以在开源社区中创建问题，而问题的参与者包括项目开发团队的核心成员或是对该问题感兴趣的外部贡献者。在开源社区中，经常会有大量的问题等待开发者进行回复和解决[1]。将推荐技术应用到开源社区的问题解决过程中，为每个问题推荐问题解决过程的参与者，能有助于提高问题解决的速度与质量，进而促进社区的协作与发展[2,3]。

近年来，许多学者提出了不同的方法，这些方法从开发者画像、开发者过去解决问题的特征，待解决问题的特征[4,5]等方面进行问题参与者的推荐。

Zhou 等人采用结构方程模型(SEM)分析发现，在开源社区中社会认同是决定开源社区用户知识贡献的主要因素[6]；Morris 等人研究发现许多问题都是被联系紧密的朋友解决的，并且联系的紧密程度影响着开发者参与问题的积极性[7,8]。所以将开发者之间的协作关系纳入推荐模型，其结果有助于提升开发者知识贡献的意愿，从而更好、更快的解决问题[9,10]，目前的推荐模型大多根据开发者的专业技能等信息进行推荐，部分工作中虽然考虑了问题提出者与开发者之间的关系，但是这些模型中考虑的是直接社交关系的影响[11,12]，没有能够反映社交关系对开发者参与积极性产生的潜在的、复杂的作用，所以本文设计了一个基于图注意力网络的问题参与者推荐模型，引入用户协作关系图以提升推荐的效果。

本文的主要工作如下：根据开源社区开发者协作的特点，将开发者协作关系信息和开发者问题参与信息与图注意力网络相结合，构造了一个双层图注意力网络模型，其中包括开

收稿日期：2022-01-13；**修回日期：**2022-03-19　**基金项目：**国家重点研发计划项目(2018YFB1003802)

作者简介：赵海燕(1975-)，女，河南人，副教授，硕导，博士，主要研究方向为服务计算、数据挖掘、推荐系统；夏文宗(1998-)(通信作者)，男，江西南昌人，硕士研究生，主要研究方向为推荐系统、机器学习、数据挖掘(wenzongxia@gmail.com)；曹健(1972-)，男，江苏人，教授，博导，博士，主要研究方向为协同计算、服务计算、网络计算、智能数据分析等；陈庆奎(1967-)，男，黑龙江人，教授，博导，博士，主要研究方向为计算机集群、并行数据库、并行理论、物联网等.

发者协作关系图和开发者问题参与图。本文在 Github 上的流行仓库[13]中选取了 7352 个问题进行了实验。实验结果表明，本文提出的 GAT-UCG 模型在推荐准确率、召回率、F-Score 三个指标上均优于基线方法。

1　相关工作

近些年来，有许多学者对于问题参与者推荐进行了研究。Jiang 等人提出了一种基于多属性的推荐评论者方法，将开发者的活跃度与文本相似度等因素纳入考虑，发现开发者的活跃度是问题参与者推荐最重要的属性[14]。Chen 等人设计了一个回答者推荐系统，有针对性的将问题推荐给有专业知识、能够回答该问题的开发者，并且他们发现，对问题进行及时的处理可以促进问题提问者与回答者之间的互动，使问题的回答者更积极地改进他的回答[15]。Davoodi 等人提出的基于混合方法的专家推荐系统，使用基于社交网络的协同过滤方法来提升参与者预测的准确率[16]。Xu 提出一种基于社交网络并使用矩阵分解技术的新推荐方法，以缓解推荐系统稀疏性的问题并提高模型在复杂内容下的准确率和多样性[17]。刘晔晖等人提出一种基于信息索引、评论网络及熵值法的混合推荐算法，其混合方法相比其独立的方法有着更好的性能[1]。

由于传统的推荐系统存在泛化能力差、表达能力不足并且难以处理非欧几里德空间数据等问题[18~20]。近些年来，能够处理非欧几里德空间数据的图神经网络在挖掘复杂网络的隐藏特征等任务上取得了巨大的成功[21]。针对基于图神经网络(GNN)的推荐算法，Kipf 等人提出了一种半监督图卷积算法(GCN)，利用图的拓扑结构和节点的信息进行标签分类[22]。Zhang 等人提出一种基于图神经网络的启发式链路预测方法，从局部子图中学习启发式，并保持着很好的泛化性能[23]。Zhang 等人提出了基于图卷积神经网络(GCN)的用户表示学习推荐方法，通过多层的图卷积层进行用户的表示学习[24]。虽然图卷积神经网络(GCN)能够很好的表示节点的特征，但 GCN 内推式(Transductive)的性质并不适合开源软件快速迭代的特点。为了提升 GCN 扩展性，Hamilton 等人提出了 GraphSAGE 模型，有效的提升了模型的灵活性和泛化能力[25]。但在实际情况中，不同的开发者对于问题的贡献程度是不同的，需要对于开发者节点之间的边给予重要性的表示。Veličković 等人提出基于注意力模型的图神经网络(GAT)，通过引入注意力机制来使模型能够根据邻居节点特征的不同来为其分配不同的权值[26]。针对图神经网络模型的改进方法，Tao 等人提出了一种基于多模型的图注意力推荐算法，使用 GNNs 从不同的模型中获取用户的偏好[27]。Fan 等人提出一种用于社交推荐的图神经网络架构(GraphRec)，使用注意力机制区分社交关系的重要性[28]。Guo 等人提出一种基于深度图神经网络的社交推荐框架(GNN-SoR),将用户与项目的特征抽象为两个图，通过图神经网络进行编码并将编码嵌入至矩阵分解的两个潜在因子中，以完成用户-项目评分矩阵中缺失的评分值[29]。Wang 等人使用不同的聚类函数来处理用户的邻居并使用注意力机制生成用户项目的表示[30]。

在本文的模型中，首先利用开源社区问题解决过程中的用户评论数据，来构建开发者的特征，之后利用图注意力机制学习开发者邻居节点的权重并进行节点特征的传播。问题的标签可以直接的体现提问者的主题和疑问所对应的模块，所以将问题标签的 One-Hot 编码作为问题节点的特征进行问题参与者的推荐。与目前的其他模型相比，本文的模型对协作关系进行了更为全面的融合。

2　基于图注意力网络的问题参与者推荐

本文首先对于开源社区中评论数据进行分析，验证了开发者之间的协作关系在问题解决的过程中起着重要的作用。为了充分的挖掘开发者协作网络，本文根据开发者之间的评论信息与开发者参与问题的记录，设计了一个双层图注意力模型(GAT-UCG)，模型的整体结构如图 1 所示。

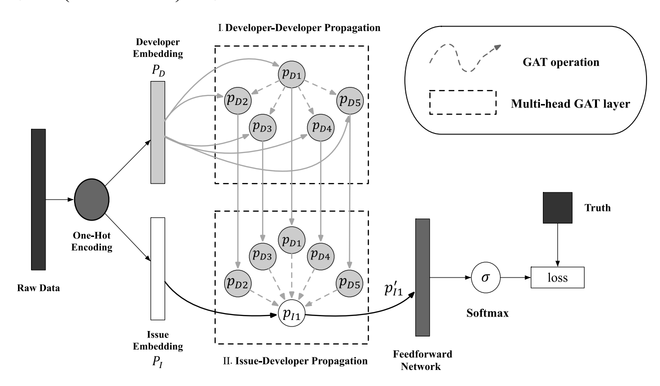


图 1　GAT-UCG 整体结构图

Fig. 1　Overall Architecture of GAT-UCG

2.1　社交关系对于开发者参与问题解决的重要性

为了验证开源社区中历史合作过的关系在新问题讨论中出现的比例，本文在 Github 上选取了两个流行仓库 tensorflow 和 kubernetes 的数据进行分析。

首先在仓库中选取特定时间段内的 Issue 作为源数据，并根据月份划定每月用户社交数据并以第一个月的数据为用户社交的基准数据，将后面月份的互动数据与基准互动数据对比，获得当前时间段里用户互动列表中历史上曾互动过的关系的比例，其中本文将曾互动过的行为定义为用户双方曾在同一个 Issue 中进行过评论行为，分析结果如表 1 所示。

表 1　tensorflow、Kubernetes 仓库中社交关系统计表

Tab. 1　Statistics of social relationships of tensorflow and Kubernetes

Repo	time bucket	interaction	interaction in historical	historical proportion
tensorflow	3/22/2020-4/30/2020	5354	\	\
	4/30/2020-5/31/2020	2434	526	21.61%
	5/31/2020-6/30/2020	1576	402	25.50%
	6/30/2020-7/31/2020	1474	438	29.71%
kubernetes	12/28/2020-1/31/2021	4724	\	\
	1/31/2021-2/28/2021	3812	1902	49.89%
	2/28/2021-3/31/2021	4756	2744	57.69%
	3/31/2021-4/30/2021	2968	1824	62.06%
	4/30/2021-5/31/2021	2482	1592	64.14%
	5/31/2021-6/30/2021	2044	1420	69.47%
	6/30/2021-7/30/2021	1278	864	67.60%

从表 1 可以看出，在 kubernetes 仓库中，在以一个月为间隔的时间段内，平均曾互动过的比例占该月总互动数目的 61.80%，在 tensorflow 社区中也存在一定比例的历史互动。分析结果表明，开发者与曾互动过的人一起参与新问题的讨论这一现象十分普遍，并且随着时间的推进，这一比例稳步升高，体现出开发者之间的协作网络逐渐扩大，所以将开发者之间的历史合作信息纳入考虑有助于提升问题参与者推荐模型的性能。

2.2　多注意力头的图注意力层

多注意力头的图注意力层作为 GAT-UCG 模型的基本单元，其基本结构如图 2 所示。该层的输入数据为一组节点特征，$P=\{\vec{p_1},\vec{p_2},\vec{p_3},\dots,\vec{p_N}\},\vec{p_i}\in\mathbb{R}^F$,其中 N 代表节点的数量，F 代表每个节点特征的数目。每一层模型的输出为一组节点的新特征，$P'=\{\vec{p_1'},\vec{p_2'},\vec{p_3'},\dots,\vec{p_N'}\},\vec{p_i'}\in\mathbb{R}^{F'}$。为了充分的将输入特征转换为高阶的嵌入表示，在每个图注意力层中，在节点的计算中引入自注意力(self-attention)机制，输入特征的计算如式(1)(2)所示。

$$e_{ij}=LeakyReLU\left(a^{T}\left[W_{p_i}\parallel W_{p_j}\right]\right) \tag{1}$$

$$\alpha_{ij}=softmax_{j}\left(e_{ij}\right)=\frac{\exp\left(e_{ij}\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(e_{ik}\right)} \tag{2}$$

其中，$\boldsymbol{P}$ 为输入的嵌入表示，$\boldsymbol{W}\in\mathbb{R}^{F'\times F}$ 为权重矩阵，$\mathcal{N}_i$ 为图 G 中节点 i 的一阶邻居。

所以，注意力层输出的嵌入表示 $\boldsymbol{P}'\in\mathbb{R}^{N*F'}$ 计算公式如式(3)所示。

$$\overrightarrow{p_i'}=\sigma\left(\sum_{j\in N_i}\alpha_{ij}W\vec{p}_j\right) \tag{3}$$

其中 i,j 属于图 G，图 G 为输入图，σ 为非线性的激活函数。

为了能够获得稳定的自注意力(self-attention)结果，采用 K 个独立的注意力头进行计算，之后将它们的计算结果进行串联，其中在最后一层，为了减少输出特征向量的维度，将拼接操作替换为平均操作，如式(4)(5)所示。

$$\overrightarrow{p_i'}=\Vert_{k=1}^{K}\sigma\left(\sum_{j\in N_i}\alpha_{ij}^{k}W^{k}\vec{p}_j\right) \tag{4}$$

$$\overrightarrow{p_i'}=\sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in N_i}\alpha_{ij}^{k}W^{k}\vec{p}_j\right) \tag{5}$$

其中，$\parallel$ 表示将特征进行串联操作，K 代表多注意力头的数目，α_{ij}^{k} 代表由第 k 个注意力头计算的注意力系数，$\boldsymbol{W}^{k}$ 代表对应输入的线性变换权重矩阵。使用多注意力机制可将注意力分配到中心节点与邻居节点之间的相关特征上，可使模型的学习能力更强。

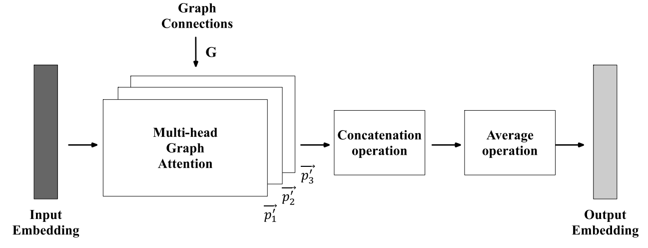


图 2　图注意力网络结构

Fig. 2　Graph attention network architecture

2.3　开发者协作图的权重传播

开发者协作图层主要负责进行开发者之间权重传播。本文将用户协作图定义为一个带权重的有向图 $G_{DC}=\langle V_D,E_D,W_D\rangle$，其中节点 V_D 表示开发者集合，节点之间的关系集合用边 E_D 表示，如果开发者 V_{Di} 至少评论或引用了开发者 V_{Dj} 提交的一个问题或回复，那么从开发者 V_{Di} 到开发者 V_{Dj} 之间就会有一条边 e_{ij}。本文通过正则表达式来提取文本中的@行为、引用他人评论等行为，获得开发者之间的协作关系。权重集合 $\boldsymbol{W}$ 反映了边的重要程度，本文通过图注意力机制来进行边的权重进行学习，使用 α_{ij} 表示 V_{Di},V_{Dj} 之间边的权重。开发者节点的嵌入表示通过该开发者历史参与问题的标签进行 Multi-hot 编码得到。开发者协作图注意力机制的输出如式(6)所示。

$$\overrightarrow{p_{Di}'}=\sigma\left(\sum_{j\in\mathcal{N}_{Di}}\frac{\exp\left(LeakyReLU\left(a^{T}\left[W_{p_{Di}}\parallel W_{p_{Dj}}\right]\right)\right)}{\sum_{k\in\mathcal{N}_{Di}}\exp\left(LeakyReLU\left(a^{T}\left[W_{p_{Di}}\parallel W_{p_{Dk}}\right]\right)\right)}W\vec{p}_{Dj}\right) \tag{6}$$

其中，$\boldsymbol{p}_{Di},\boldsymbol{p}_{Dj}$ 分别表示开发者节点 i,j 的嵌入表示，$\mathcal{N}_{Di}$ 为开发者节点 i 的一阶邻居节点。

图 3 显示了开发者协作图中的一部分示例，如开发者 p_{D4} 与开发者 p_{D5} 存在协作关系，因此 p_{D4} 到 p_{D5} 有一条边,其权重为 α_{45}。

2.4　开发者-问题参与图的权重传播

本文将开发者-问题参与图定义为 $G_{ID}=\langle V_I,V_D,E_{I-D},W_{I-D}\rangle$，该图为二分图。其中顶点 V_I 表示问题集合，顶点 V_D 表示开发者集合，问题节点与开发者节点之间的关系集合用边 E_{I-D} 表示，如果开发者 V_{Di} 至少评论了问题 V_{Ij} 一次，那么从 V_{Di} 到 V_{Ij} 就会有一条边 e_{ij}。权重集合 $\boldsymbol{W}$ 反映了边的重要程度，本文通过注意力机制来对开发者与问题之间边的权重进行学习,使用 α_{ij} 表示 V_{Di},V_{Ij} 之间边的权重大小。开发者-问题参与图注意力机制的输出如式(7)所示。

$$\overrightarrow{p_i'}=\sigma\left(\sum_{D\in\mathcal{N}_d}\frac{\exp\left(LeakyReLU\left(a^{T}\left[W_{p_i}\parallel W_{p_D}\right]\right)\right)}{\sum_{k\in\mathcal{N}_d}\exp\left(LeakyReLU\left(a^{T}\left[W_{p_i}\parallel W_{p_k}\right]\right)\right)}W\vec{p}_D\right) \tag{7}$$

其中，$\boldsymbol{p}_D$ 为开发者节点的嵌入表示，$\boldsymbol{p}_i$ 为问题节点的嵌入表示，$\mathcal{N}_d$ 为问题节点的一阶邻居节点。

图 4 显示了开发者-问题参与图中的一部分示例，如开发者 p_{D1},p_{D2},p_{D3} 参与了问题 p_{I1} 的讨论，因此 p_{D1},p_{D2},p_{D3} 到 p_{I1} 分别有一条边，其分别为 $\alpha_{11},\alpha_{21},\alpha_{31}$。

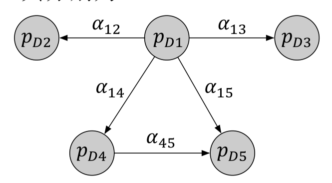


图 3　开发者协作关系图示例

Fig. 3　An example of developer cooperation graph

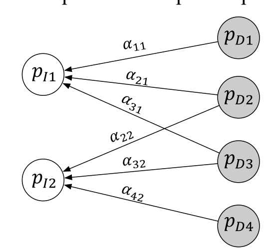


图 4　开发者-问题参与图示例

Fig. 4　An example of Developer-Issue participation graph

2.5　模型预测和优化

为了学习模型参数，使模型可以更好的建模问题与开发者之间的特征。本文采用 Log*Softmax* 进行问题解决参与者的预测，预测层输出如式(8)所示。

$$p_i=\log\left(\frac{\exp\left(a_i\right)}{\sum_{k=1}^{C}\exp\left(a_k\right)}\right) \tag{8}$$

其中，$\boldsymbol{a}_1,\boldsymbol{a}_2,\boldsymbol{a}_3,\ldots,\boldsymbol{a}_C$ 是问题节点 I 的特征，p_i 为该问题节点 I 应该被分配给开发者 i 的概率，通过上式可以获得该问题节点 I 分配给每个开发者的概率，根据概率的高低进行排序，推荐 Top-N 的问题解决参与者。使用 Log*Softmax* 能够加快模型运算的速度，提高数据的稳定性。模型采用梯度下降法训练更新模型参数。

3　数据及实验

3.1　问题及用户评论数据

本文在 Github 上选择了两个流行仓库(tensorflow、kubernetes)作为实验数据。通过 Github API 获得了仓库特定时间段内的 Issue 信息，包含 Issue 的标签以及所有参与该问题的用户信息，共获取了 7352 条 Issue 以及 58814 条评论行为，数据集见表 2。

表 2　tensorflow、kubernetes 数据集

Tab. 2　Datasets of tensorflow and kubernetes

Repo	issue	comment	developer	label attributes	time bucket
tensorflow	2459	13067	497	79	3/22/2020-7/28/2020
kubernetes	4893	45747	896	121	12/28/2020-7/30/2021

3.2　评测指标

为了验证算法的性能，采用召回率、精确率和 F-Score 作为评价的指标，推荐结果的召回率计算方式如式(9)所示。

$$Recall=\frac{\sum_{i=1}^{|Issue_s|}|R(i)\cap T(i)|}{\sum_{i=1}^{|Issue_s|}|T(i)|} \tag{9}$$

推荐结果的精确率计算方式如式(10)所示。

$$Precision=\frac{\sum_{i=1}^{|Issue_s|}|R(i)\cap T(i)|}{\sum_{i=1}^{|Issue_s|}|R(i)|} \tag{10}$$

推荐结果的 F-score 值计算方式如式(11)所示。

$$F-score=2*\frac{precision*recall}{precision+recall} \tag{11}$$

其中 $Issue_s$ 表示问题的测试集，$R(i)$ 表示根据开发者在训练集上的行为作出的 Top-N 推荐列表，$T(i)$ 表示实际参与问题的用户列表。

3.3　实验设置

本文在 Python3.8，Pytorch1.9.0，Cuda11.4，RTX3090 环境下完成 GAT-UCG 与基线算法的对比实验。在实验中，按照 8:1:1 的比例划分数据集来构造训练集、验证集和测试集；GAT-UCG 模型超参数的设置如表 3 所示。将模型默认的训练轮次(epochs)设置为 2000，并且当验证集上的评测指标在 100 轮内没有变化时提前结束训练。基线方法 GAT 和 GCN 中的嵌入维度、学习率、批处理大小和注意力头数目与 GAT-UCG 模型相同，其他超参数默认与原始论文或代码一致。实验目标是为测试集中的每个问题推荐排名靠前的 N 个开发者，并使用 Recall@N 、Precision@N、F-Score@N 三个指标来衡量各模型的性能。

表 3　模型的超参数设置

Tab. 3　Hyperparameter setting of model

超参数名称	参数值
问题、开发者嵌入维度	237(tensorflow 社区)/363(kubernetes 社区)
学习率	0.005
隐藏层数目	8
注意力头数目	8
Drop out Rate	0.6
L2 正则项系数	5e-4

3.4　消融实验

为了评估图注意力层中多注意力头的有效性以及贡献，本节进行了消融实验分析了单注意力头与多注意力头。根据统计，在本文选取的两个流行仓库中，99.79%的问题参与人数在 10 位及以下，所以将推荐的列表长度限制在 10 人以内进行 Top-N 的问题参与者推荐的召回率评估，结果如表 4 所示。

为了更加清晰的显示性能的差别，将表 4 转换成图 5 的形式。消融实验结果显示，基于多注意力头的图注意力网络模型相比基于单注意力头的图注意力网络模型在 tensorflow 和 kubernetes 两个仓库的问题参与者 Top-N 的推荐召回率平均提高 8.57%，表明基于多注意力头的图注意力模型相比基于单注意力头的图注意力网络模型能稳定传播邻居节点信息并且提升模型的推荐性能。

表 4　不同注意力头数 Top-N 推荐的召回率

Tab. 4　Recall rate of Top-N recommendations with different numbers of heads

Repo	attention head	2	4	6	8	10
tensorflow	Multi	39.67%	48.29%	52.85%	55.59%	58.89%
	Single	27.66%	39.07%	44.43%	46.82%	51.78%
kubernetes	Multi	45.01%	56.16%	61.18%	64.05%	66.04%
	Single	34.55%	46.35%	53.29%	57.29%	60.77%

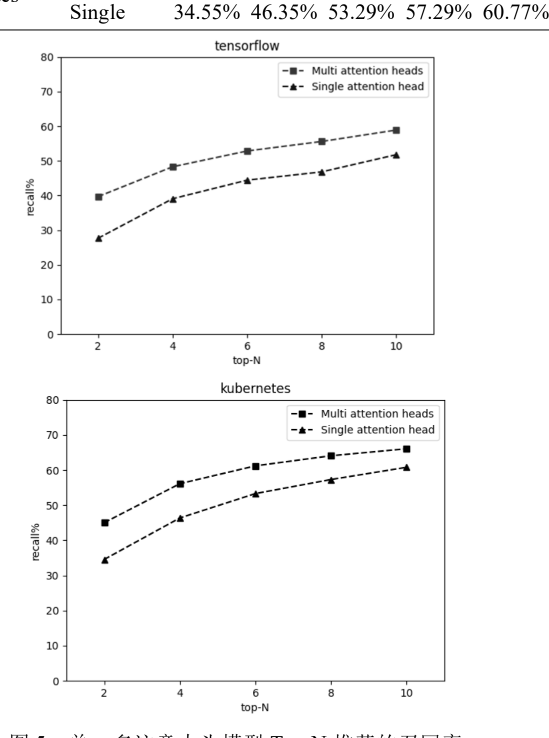


图 5　单、多注意力头模型 Top-N 推荐的召回率

Fig. 5　Recall rate of Top-N recommendations of single and multi-head attention models

3.5　实验结果

根据消融实验的结果，本文将多注意力头的机制应用到 GAT-UCG 模型中，本文比较了不同的方法在推荐 Top-2、Top-4、Top-6、Top-8、Top-10 个参与者时的推荐性能，实验结果如表 5~7 所示。

在表 5~7 中，第一行是本文所提出的基于开发者协作图和问题参与图的双层图注意力网络(GAT-UCG)模型。第二行是图注意力网络(GAT)模型[26]。第三行是图卷积神经网络(GCN)模型[22]。第四行是基于开发者社交网络的用户画像模型(SN)[1]。第五行是基于矩阵分解的协同过滤模型(MF)[17]。

表 5　不同算法 Top-N 推荐的召回率

Tab. 5　Recall rate of Top-N recommendations of different methods

Repo	baseline	2	4	6	8	10
tensorflow	GAT-UCG	39.67%	48.29%	52.85%	55.59%	58.89%
	GAT	16.14%	28.32%	36.80%	43.65%	45.92%
	GCN	11.63%	20.68%	28.28%	33.36%	36.68%
	基于社交网络的用户画像(SN)	22.69%	26.10%	28.93%	30.54%	31.39%
	基于矩阵分解的协同过滤算法(MF)	5.11%	14.82%	21.45%	24.82%	25.52%
kubernetes	GAT-UCG	45.01%	56.16%	61.18%	64.05%	66.04%
	GAT	28.70%	39.59%	45.27%	49.10%	51.27%
	GCN	12.18%	22.17%	29.11%	35.21%	37.27%
	基于社交网络的用户画像(SN)	18.04%	21.08%	23.41%	25.58%	26.22%
	基于矩阵分解的协同过滤算法(MF)	4.22%	8.31%	12.08%	14.98%	17.13%

表 6　不同算法 Top-N 推荐的精确率

Tab. 6　Precision rate of Top-N recommendations of different methods

Repo	baseline	2	4	6	8	10
tensorflow	GAT-UCG	39.80%	24.22%	17.62%	13.94%	11.42%
	GAT	16.20%	14.23%	12.30%	10.95%	9.21%
	GCN	11.67%	10.37%	9.45%	8.36%	7.36%
	基于社交网络的用户画像(SN)	25.39%	14.07%	9.90%	7.57%	6.10%
	基于矩阵分解的协同过滤算法(MF)	12.84%	12.56%	10.78%	8.93%	8.87%
kubernetes	GAT-UCG	53.05%	33.10%	23.86%	18.71%	15.42%
	GAT	33.30%	22.97%	17.51%	14.24%	11.90%
	GCN	14.13%	12.86%	11.26%	10.21%	8.65%
	基于社交网络的用户画像(SN)	18.92%	12.30%	8.61%	6.74%	4.74%
	基于矩阵分解的协同过滤算法(MF)	8.41%	8.21%	7.95%	7.42%	6.64%

表 7　不同算法 Top-N 推荐的 F-score

Tab. 7　F-scores of Top-N recommendations of different methods

Repo	baseline	2	4	6	8	10
tensorflow	GAT-UCG	39.73%	32.26%	26.43%	22.30%	19.02%
	GAT	16.17%	18.94%	18.43%	17.51%	15.34%
	GCN	11.64%	13.81%	14.16%	13.36%	12.26%
	基于社交网络的用户画像(SN)	23.96%	18.28%	14.75%	12.13%	10.22%
	基于矩阵分解的协同过滤算法(MF)	7.31%	13.59%	14.34%	13.13%	13.16%
kubernetes	GAT-UCG	48.70%	41.65%	34.33%	28.96%	25.01%
	GAT	30.82%	29.25%	25.25%	22.08%	19.32%
	GCN	13.08%	16.27%	16.24%	15.82%	14.04%
	基于社交网络的用户画像(SN)	18.47%	15.54%	12.59%	10.67%	8.03%
	基于矩阵分解的协同过滤算法(MF)	5.62%	8.26%	9.59%	9.92%	9.57%

3.6　结果分析

为了直观地表示结果，分别将表 5~7 转换成图 6~8 的形式。实验结果显示，本文所提出的 GAT-UCG 推荐模型在实验的两个仓库上的推荐效果都比对比的四个算法的效果要好。其中，在 tensorflow 仓库上，本文所提出的方法召回率最多提升了 39.61%，在 kubernetes 仓库上，召回率最多提升了 54.61%。

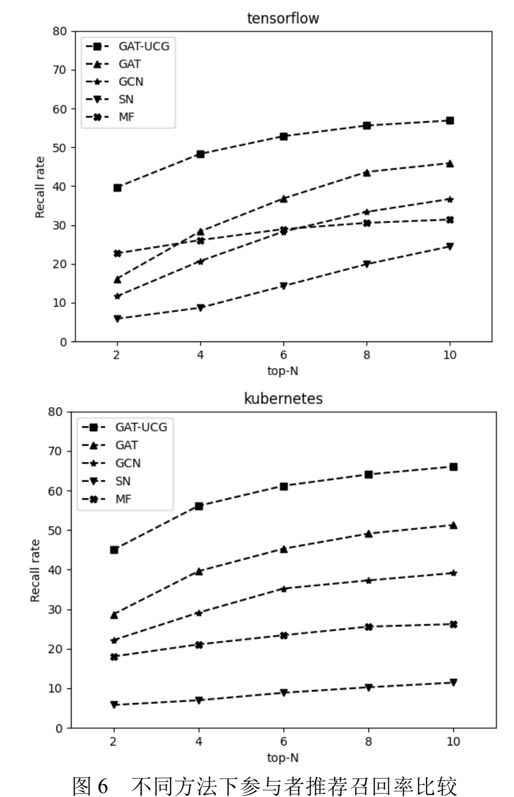


图 6　不同方法下参与者推荐召回率比较

Fig. 6　Comparisons of recall rate of different methods for participant recommendation

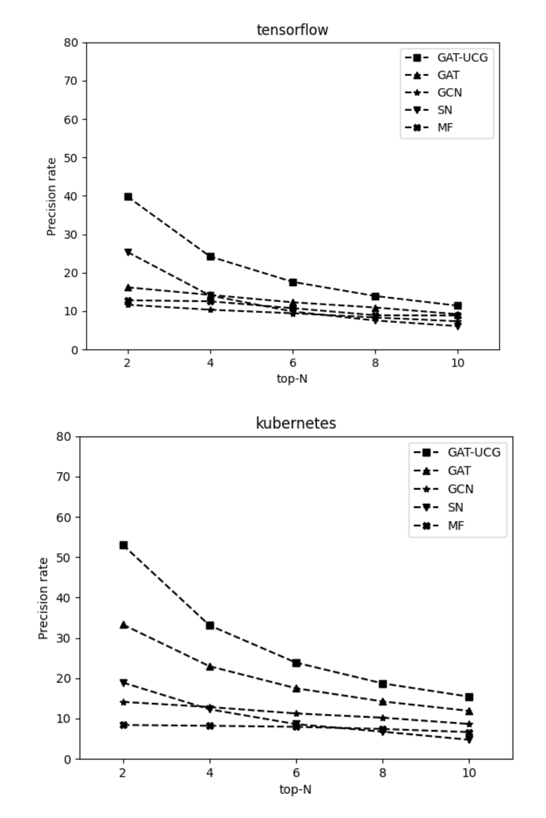


图 7　不同方法下参与者推荐精确率比较

Fig. 7　Comparisons of precision rate of different methods for participant recommendation

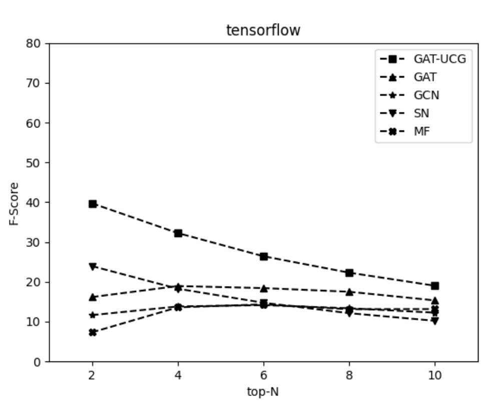
比较所提出的模型和数据集的不同结果，有以下的观察和结论：

(1)基于图神经网络的模型在大部分实验结果上要优于基于矩阵分解和基于社交网络的传统模型。这说明图神经网络模型能够很好的挖掘开发者协作关系和兴趣偏好的特征。

(2)在基于图神经网络的模型中，将开发者协作信息与开发者-Issue 参与信息进行综合考虑相比于只利用开发者-Issue 参与信息的模型可以获得更好的问题参与者推荐性能。

(3)比较两个实验数据集上的实验效果，图神经网络模型在 kubernetes 仓库的数据上收到了最优的效果，kubernetes 仓库有更多的用户数目、互动数和标签特征数，这表明图的内容越丰富，嵌入向量的信息越多，实验效果越好。因此，当将所提出的模型应用于更大的数据集时，其效果更加明显。

(4)基于社交网络的用户画像模型在 tensorflow 仓库上 Top-2 的推荐效果相比于基于图注意力网络的推荐效果更好，是由于 tensorflow 仓库参与人员更为固定并且标签类型更少导致基于社交网络能在推荐人数较少的情况下效果更好。此情况在参与人数更多、标签更丰富的 kubernetes 仓库中并不明显。实验结果显示，在 tensorflow 仓库中，从 Top-4 开始，图注意力网络模型的推荐性能开始超过基于社交网络的用户画像模型，说明图神经网络模型能更好的适应复杂的情况。



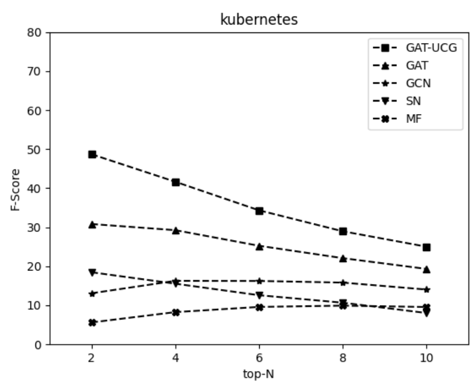


图 8　不同方法下参与者推荐 F-score 比较

Fig. 8　Comparisons of F-scores of different methods for participant recommendation

4　结束语

本文通过深入分析已有的问题参与者推荐方法，提出了一种基于图注意力网络的问题解决参与者推荐模型并详细描述了设计的模型。本文选取了 Github 上流行仓库的数据进行实验，使用 Precision、Recall 和 F-Score 作为模型的评估指标，将本文所提出的推荐模型与 GAT、GCN、SN 和 MF 进行对比，实验结果表明本文提出的推荐模型在数据集上的指标均优于基线算法，可以有效的推荐问题参与者。同时，本文的模型引入了开发者之间的协作关系，可以更好的根据开发者的社交关系进行推荐。

本文的工作还有不少可拓展的地方，例如在本文中选取问题的标签作为问题的描述特征，然而，在实际中还有许多问题来不及打标签或者标签不全等问题。在未来的工作中，可以考虑将更多的问题特征纳入考虑，以提高问题参与者推荐模型的性能。

参考文献：

[1] 刘晔晖，赵海燕，曹健，陈庆奎．开源社区中 Issue 解决过程的参与者推荐方法 [J]．小型微型计算机系统，2020, 41 (09): 1930-1934. (Liu Ye-hui, Zhao Hai-yan, Cao Jian, *et al.* Participants recommendation approaches for issue solving process in open source community [J]. Journal of Chinese Computer Systems, 2020, 41 (9): 1930-1934.)

[2] MARLOW J, DABBISH L, HERBSLEB J. Impression formation in online peer production: Activity traces and personal profiles in github [C]// Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW'13) . New York, NY, USA, 2013: 117-128.

[3] TSAY J, DABBISH L, HERBSLEB J. Influence of social andtechnical factors for evaluating contribution in github [C]// Proceedings of the 36th International Conference on Software Engineering (ICSE 2014) . New York, NY, USA, 2014: 356-366.

[4] Yeh R T, Ramamoorthy C V. Proceedings of the 2nd international conference on softw are engineering [J]. Scientific American, 1976, 172 (4): 297-301.

[5] Montandon J E, Silva L L, Valente M T. Identifying experts in softw are libraries and framew orks among Git Hub users [C]// Proceedings of the 16th International Conference on M ining Softw are Repositories, IEEE Press, 2019: 276-287.

[6] 周涛，王超．开源软件社区用户知识贡献行为研究 [J]．科研管理，2020, 41 (02): 202-209. (Zhou Tao, Wang Chao. A research on knowledge contribution behaviors of open source software community users [J]. Science Research Management, 2020, 41 (02): 202-209.)

[7] Morris M R, Teevan J, Panovich K. A comparison of information seeking using search engines and social networks [C]// Fourth International AAAI Conference on Weblogs and Social Media. 2010.

[8] White R W, Richardson M, Liu Y. Effects of community size and contact rate in synchronous social Q&A [C]// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2011: 2837-2846.

[9] Alami A, Dittrich Y, Wasowski A. Influencers of quality assurance in an open source community [C]// 2018 IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE) . IEEE, 2018: 61-68.

[10] SOWE S K, STAMELOS I, ANGELIS L. Understandin g know-ledge sharing activities in free/open source software projects: An empirical study [J]. Journal of Systems and Software, 2008, 3 (81): 431-446.

[11] Wang H, Wang T, Yin G, *et al.* Linking issue tracker with Q&Asites for know ledge sharing across communities [J]. IEEE Transactions on Services Computing, 2015, 11 (5): 782-795.

[12] Guo L, Chen Q, Han W, *et al.* Collaborative topic prediction model for user interest recommendation in online social netw orks [J]. International Journal of Digital Content Technology&its Applic, 2012, 6 (23): 62-73.

[13] 王伟，周添一，赵生宇，范家宽．全球开源生态发展现状研究 [J]．信息通信技术与政策, 2020 (05): 38-44. (Wang Wei, Zhou Tianyi, Zhao Shengyu, Fan Jiakuan. Research on the development of global open source ecology [J]. Science Research Management, 2020, 41 (02): 202-209.)

[14] Jiang J, Yang Y, He J, *et al.* Who should comment on this pull request?analyzing attributes for more accurate commenter recommendation in pull-based development [J]. Information and Software Technology, 2017, 84: 48-62.

[15] Chen T, Cai J, Wang H, *et al.* Instant expert hunting: building an answerer recommender system for a large scale Q&A website [C]// Proceedings of the 29th Annual ACM Symposium on Applied Computing. 2014: 260-265.

[16] Davoodi E, Afsharchi M, Kianmehr K. A social network-based approach

to expert recommendation system [C]// International conference on hybrid artificial intelligence systems. Springer, Berlin, Heidelberg, 2012: 91-102.

[17] Xu C. A novel recommendation method based on social network using matrix factorization technique [J]. Information processing & management, 2018, 54 (3): 463-474.

[18] 程龙，李涵. 基于矩阵分解的推荐算法研究综述 [J]. 北京信息科技大学学报（自然科学版）, 2021, 36 (02): 38-45+51. (Cheng Long, Li Han. A review of recommendation algorithms based on matrix factorization [J]. Journal of Beijing Information Science & Technology University, 2021, 36 (02): 38-45+51.)

[19] 秦川，祝恒书，庄福振，等. 基于知识图谱的推荐系统研究综述 [J]. 中国科学：信息科学, 2020, 50 (07): 937-956. (Qin Chuan, Zhu Hengshu, Zhuang Fuzhen, et. A survey on knowledge graph-based recommender systems [J]. Scientia Sinica (Informationis) , 2020, 50 (07): 937-956.)

[20] 周万珍，曹迪，许云峰，刘滨. 推荐系统研究综述 [J]. 河北科技大学学报, 2020, 41 (01): 76-87. (Zhou Wanzhen, Cao Di, Xu Yunfeng, Liu Bin. A survey of recommendation systems [J]. Journal of Hebei University of Science and Technology, 2020, 41 (01): 76-87. s)

[21] Wu Z, Pan S, Chen F, *et al.* A comprehensive survey on graph neural networks [J]. IEEE transactions on neural networks and learning systems, 2020, 32 (1): 4-24.

[22] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [J]. arXiv preprint arXiv: 1609. 02907, 2016.

[23] Zhang M, Chen Y. Link prediction based on graph neural networks [J]. Advances in Neural Information Processing Systems, 2018, 31: 5165-5175.

[24] Zhang S, Yin H, Chen T, *et al.* Gcn-based user representation learning for unifying robust recommendation and fraudster detection [C]// Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 689-698.

[25] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 1025-1035.

[26] Veličković P, Cucurull G, Casanova A, *et al.* Graph attention networks [J]. arXiv preprint arXiv: 1710. 10903, 2017.

[27] Tao Z, Wei Y, Wang X, *et al.* MGAT: multimodal graph attention network for recommendation [J]. Information Processing & Management, 2020, 57 (5): 102277.

[28] Fan W, Ma Y, Li Q, *et al.* Graph neural networks for social recommendation [C]// The World Wide Web Conference. 2019: 417-426.

[29] Guo Z, Wang H. A deep graph neural network-based mechanism for social recommendations [J]. IEEE Transactions on Industrial Informatics, 2020, 17 (4): 2776-2783.

[30] Wang J, Xie H, Wang F L, *et al.* Top-N personalized recommendation with graph neural networks in MOOCs [J]. Computers and Education: Artificial Intelligence, 2021, 2: 10001